

FILEID**INVOL

N 1

IIIIII NN NN II IIII VV VV 000000 LL
IIIIII NN NN II IIII VV VV 000000 LL
IIII NN NN II IIII VV VV 00 00 LL
IIII NNNN NN II IIII VV VV 00 00 LL
IIII NNNN NN II IIII VV VV 00 00 LL
IIII NN NN NN II IIII VV VV 00 00 LL
IIII NN NN NN II IIII VV VV 00 00 LL
IIII NN NNNN NN II IIII VV VV 00 00 LL
IIII NN NNNN NN II IIII VV VV 00 00 LL
IIII NN NN NN II IIII VV VV 00 00 LL
IIII NN NN NN II IIII VV VV 00 00 LL
IIII NN NN NN II IIII VV VV 00 00 LL
IIII NN NN NN II IIII VV VV 00 00 LL
IIII NN NN NN II IIII VV VV 00 00 LL
IIII NN NN NN II IIII VV VV 00 00 LL
IIII NN NN NN II IIII VV VV 00 00 LL
IIII NN NN NN II IIII VV VV 00 00 LL
IIII NN NN NN II IIII VV VV 00 00 LL
IIII NN NN NN II IIII VV VV 000000 LL
IIII NN NN NN II IIII VV VV 000000 LL
....
IIIIII NN NN II IIII VV VV 000000 LLLLLLLL
IIIIII NN NN II IIII VV VV 000000 LLLLLLLL
....

LL II IIII SSSSSSS
LL II IIII SSSSSSS
LL II IIII SS SS
LL II IIII SS SS
LL II IIII SSSSSS
LL II IIII SSSSSS
LL II IIII SS
LL II IIII SS
LL II IIII SS
LL II IIII SS
LL LLLLLLLL II IIII SSSSSSS
LL LLLLLLLL II IIII SSSSSSS

```
1 0001 0 MODULE INIVOL (
2 0002 0   MAIN = INIT VOLUME,
3 0003 0   LANGUAGE (B[ISS32]),
4 0004 0   IDENT = 'V04-000'
5 0005 0   ) =
6 0006 1 BEGIN
7
8 0008 1
9 0009 1 ****
10 0010 1
11 0011 1   * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
12 0012 1   * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
13 0013 1   * ALL RIGHTS RESERVED.
14 0014 1
15 0015 1   * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
16 0016 1   * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
17 0017 1   * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
18 0018 1   * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
19 0019 1   * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
20 0020 1   * TRANSFERRED.
21 0021 1
22 0022 1   * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
23 0023 1   * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
24 0024 1   * CORPORATION.
25 0025 1
26 0026 1   * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
27 0027 1   * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
28 0028 1
29 0029 1
30 0030 1 ****
31 0031 1
32 0032 1 ++
33 0033 1
34 0034 1   FACILITY: INIT Utility Structure Level 2
35 0035 1
36 0036 1   ABSTRACT:
37 0037 1
38 0038 1   This is the main program for the INITIALIZE volume utility.
39 0039 1
40 0040 1   ENVIRONMENT:
41 0041 1
42 0042 1   STARLET operating system, including privileged system services
43 0043 1   and internal exec routines.
44 0044 1
45 0045 1 --
46 0046 1
47 0047 1
48 0048 1   AUTHOR: Andrew C. Goldstein, CREATION DATE: 9-Nov-1977 18:17
49 0049 1
50 0050 1   MODIFIED BY:
51 0051 1
52 0052 1   V03-003 HH0041      Hai Huang      24-Jul-1984
53 0053 1   Remove REQUIRE 'LIBDS:[VMSLIB.OBJ]MOUNTMSG.REQ'.
54 0054 1
55 0055 1   V03-002 MCN0140      Maria del C. Nasr      29-Nov-1983
56 0056 1   Changes required by new CLI interface.
57 0057 1
```

```

58 0058 1 V03-001 STJ3038 Steven T. Jeffreys, 14-Oct-1982
59 0059 1 If the INIT fails, free up the drive with an IOS_AVAILABLE i/o.
60 0060 1
61 0061 1 V0103 ACG0072 Andrew C. Goldstein, 15-Oct-1979 16:03
62 0062 1 Check both primary and secondary device char
63 0063 1
64 0064 1 V0102 ACG0069 Andrew C. Goldstein, 9-Oct-1979 16:31
65 0065 1 Remove device data table
66 0066 1
67 0067 1 V0101 ACG25501 Andrew C. Goldstein, 21-Aug-1979 11:34
68 0068 1 Clear volume valid only if volume was not mounted
69 0069 1
70 0070 1 V0100 ACG00001 Andrew C. Goldstein, 10-Oct-1978 21:28
71 0071 1 Previous revision history moved to [INIT.SRC]INIT.REV
72 0072 1 !**
73 0073 1
74 0074 1
75 0075 1 LIBRARY 'SYSSLIBRARY:LIB.L32';
76 0076 1 REQUIRE 'SRCS:INIDEF.B32';
77 0367 1 REQUIRE 'LIBDS:[VMSLIB.OBJ]INITMSG.B32';
78 0499 1
79 0500 1
80 0501 1 FORWARD ROUTINE
81 0502 1 INIT_VOLUME, ! main program
82 0503 1 MAIN_HANDLER; ! condition handler

```

```
84 0504 1 !+
85 0505 1 !-
86 0506 1 ! General impure storage for the INIT utility.
87 0507 1 !-
88 0508 1 !-
89 0509 1 !-
90 0510 1 GLOBAL LITERAL
91 0511 1 NAMEBUF_LEN = 32; ! length of device name buffer
92 0512 1 USERNAME_LEN = 12; ! maximum length of user name string
93 0513 1 !-
94 0514 1 GLOBAL
95 0515 1 CLEANUP_FLAGS : BITVECTOR [32], ! cleanup action flags
96 0516 1 CHANNEL, ! channel for all I/O
97 0517 1 PROCESS_UIC, ! UIC of this process
98 0518 1 DEVICE_CHAR : BBLOCK [DIBSK_LENGTH], ! buffer for device characteristics
99 0519 1 !-
100 0520 1 DEVICE_CHAR2 : BBLOCK [DIBSK_LENGTH], ! buffer for sec. device characteristics
101 0521 1 !-
102 0522 1 PHYS_NAME : VECTOR [2], ! physical device name descriptor
103 0523 1 NAME_BUFFER : VECTOR [NAMEBUF_LEN, BYTE], ! buffer for physical device name
104 0524 1 !-
105 0525 1 USER_STRING : VECTOR [USERNAME_LEN, BYTE]; ! buffer for user name string
106 0526 1 !-
107 0527 1 !-
108 0528 1 GLOBAL
109 0529 1 DEVCHAR_DESC : VECTOR [2] INITIAL (DIBSK_LENGTH, DEVICE_CHAR), ! descriptor for device characteristics
110 0530 1 !-
111 0531 1 DEVCHAR_DESC2 : VECTOR [2] INITIAL (DIBSK_LENGTH, DEVICE_CHAR2); ! descriptor for sec. device characteristics
112 0532 1 !-
```

```
114 0533 1 GLOBAL ROUTINE INIT_VOLUME =
115 0534 1
116 0535 1 !++
117 0536 1
118 0537 1 FUNCTIONAL DESCRIPTION:
119 0538 1
120 0539 1 This is the main routine of the INITIALIZE volume utility.
121 0540 1
122 0541 1
123 0542 1 CALLING SEQUENCE:
124 0543 1 INIT_VOLUME (ARG1, ARG2)
125 0544 1
126 0545 1 INPUT PARAMETERS:
127 0546 1 ARG1: program start address
128 0547 1 ARG2: CLI service callback address
129 0548 1
130 0549 1 IMPLICIT INPUTS:
131 0550 1 NONE
132 0551 1
133 0552 1 OUTPUT PARAMETERS:
134 0553 1 NONE
135 0554 1
136 0555 1 IMPLICIT OUTPUTS:
137 0556 1 NONE
138 0557 1
139 0558 1 ROUTINE VALUE:
140 0559 1 assorted status values
141 0560 1
142 0561 1 SIDE EFFECTS:
143 0562 1 volume initialized
144 0563 1
145 0564 1 !--
146 0565 1
147 0566 2 BEGIN
148 0567 2
149 0568 2 LOCAL
150 0569 2 P,
151 0570 2 STATUS,                                ! string scan pointer
152 0571 2 DEVICE_NAME : BBLOCK [DSC$C_S_BLN]; ! system service status
153 0572 2
154 0573 2 EXTERNAL
155 0574 2 INIT_OPTIONS : BITVECTOR,             ! device name descriptor
156 0575 2 DEVICE_STRING : BBLOCK [DSC$C_S_BLN], ! parser option flags
157 0576 2 USER_NAME : BBLOCK [DSC$C_S_BLN];   ! device name string descriptor
158 0577 2
159 0578 2 EXTERNAL ROUTINE
160 0579 2 INIT_PARSE,                          ! user name string descriptor
161 0580 2 TRAN_LOGNAME,                        ! command parser
162 0581 2 INIT_TAPE,                           ! translate logical name
163 0582 2 INIT_DISK;                           ! initialize magtape
164 0583 2
165 0584 2 BIND OPTIONS = INIT_OPTIONS : VECTOR;
166 0585 2
167 0586 2 ENABLE MAIN_HANDLER;
168 0587 2
169 0588 2 ! Get the UIC of this process for use in defaulting volume ownership.
170 0589 2
```

```
171 0590 2
172 P 0591 2 $GETJPI (ITMLST = UPLIT (WORD (4), WORD (JPIS_UIC),
173 0592 2 LONG (PROCESS_UIC, 0, 0)));
174 0593 2
175 P 0594 2 USER_NAME [DSC$A_POINTER] = USER_STRING;
176 0595 2 $GETJPI (ITMLST = UPLIT (WORD (USERNAME_LEN), WORD (JPIS_USERNAME),
177 0596 2 LONG (USER_STRING, USER_NAME[DSC$W_LENGTH], 0)));
178 0597 2
179 0598 2 ! Parse the command line. Errors are signalled.
180 0599 2 !
181 0600 2
182 0601 2 CHANNEL = 0;
183 0602 2
184 0603 2 INIT_PARSE ();
185 0604 2
186 0605 2 ! First allocate the volume and assign a channel to it. Allocating does most
187 0606 2 of the checking that we in fact have a right to touch this device.
188 0607 2 Note that we append a zero to the device name if a unit-number is not
189 0608 2 present. This prevents a generic device search, which is not desirable
190 0609 2 in doing an INIT.
191 0610 2 !
192 0611 2
193 0612 2 CH$FILL (0, DSC$C_S_BLN, DEVICE_NAME );
194 0613 2 DEVICE_NAME [DSC$B_C[ASS] = DSC$R_CLASS_D;
195 0614 2 DEVICE_NAME [DSC$W_LENGTH] = NAMEBUF_LEN;
196 0615 2 DEVICE_NAME [DSC$A_POINTER] = NAME_BUFFER;
197 0616 2
198 0617 2 TRAN_LOGNAME (DEVICE_STRING [DSC$W_LENGTH], DEVICE_NAME [DSC$W_LENGTH]);
199 0618 2
200 0619 2 P = CH$FIND_CH (.DEVICE_NAME [DSC$W_LENGTH], .DEVICE_NAME [DSC$A_POINTER], ':');
201 0620 2
202 0621 2 ! Use the low word, since high word has descriptor information.
203 0622 2
204 0623 2 IF CH$FAIL (.P)
205 0624 2 THEN
206 0625 2 P = .DEVICE_NAME [DSC$W_LENGTH] + .DEVICE_NAME [DSC$A_POINTER];
207 0626 2
208 0627 2 IF CH$RCHAR (.P-1) LSSU '0' OR CH$RCHAR (.P-1) GTRU '9'
209 0628 2 THEN
210 0629 3 BEGIN
211 0630 3 CH$WCHAR_A ('0', P);
212 0631 3 CH$WCHAR_A (':', P);
213 0632 3 DEVICE_NAME [DSC$W_LENGTH] = .DEVICE_NAME [DSC$W_LENGTH] + 1;
214 0633 2 END;
215 0634 2
216 0635 2 PHYS_NAME[0] = NAMEBUF_LEN;
217 0636 2 PHYS_NAME[1] = NAME_BUFFER;
218 P 0637 2 STATUS = $ALLOC (DEVNAM = DEVICE_NAME [DSC$W_LENGTH],
219 0638 2 PHYLEN = PHYS_NAME[0],
220 0639 2 PHYBUF = PHYS_NAME[0]);
221 0640 2 IF NOT .STATUS
222 0641 2 THEN
223 0642 2 ERR_EXIT (.STATUS);
224 0643 2
225 P 0644 2 STATUS = $ASSIGN (DEVNAM = PHYS_NAME[0],
226 0645 2 CHAN = CHANNEL);
227 0646 2 IF NOT .STATUS
```

```
228 0647 2 THEN
229 0648 2   ERR_EXIT (.STATUS);
230 0649 2
231 0650 2 ! Get the device characteristics and make sure that the device can be
232 0651 2 initialized - i.e., that it is file oriented, etc. A mismatch between
233 0652 2 primary and secondary device characteristics indicates a spooled device
234 0653 2 or something else strange - reject it if so.
235 0654 2
236 0655 2
237 0656 2 $GETCHN (CHAN = .CHANNEL, PRIBUF = DEVCHAR_DESC, SCDBUF = DEVCHAR_DESC2);
238 0657 2
239 0658 2 IF CHSNEQ (DIBSK_LENGTH, DEVICE_CHAR, DIBSK_LENGTH, DEVICE_CHAR2, 0)
240 0659 2 OR NOT .DEVICE_CHAR[DEV$V_FOD]
241 0660 2 THEN ERR_EXIT (SSS_NOTFILEDEV);
242 0661 2
243 0662 2 IF NOT .DEVICE_CHAR[DEV$V_AVL]
244 0663 2 THEN ERR_EXIT (SSS_DEVOFF[INE]);
245 0664 2
246 0665 2 IF .DEVICE_CHAR[DEV$V_MNT]
247 0666 2 THEN ERR_EXIT (SSS_DEVOUNT);
248 0667 2
249 0668 2 ! Device is now known not to be mounted.
250 0669 2
251 0670 2
252 0671 2 CLEANUP_FLAGS[CLF_CLEARVALID] = 1;
253 0672 2
254 0673 2 IF
255 0674 3   BEGIN
256 0675 3     IF .DEVICE_CHAR[DEV$V_SQD]
257 0676 3     THEN
258 0677 4       ((.OPTIONS[0] AND NOT TAPE_OPTIONS) NEQ 0
259 0678 4       OR (.OPTIONS[1] AND NOT TAPE_OPTIONS2) NEQ 0 )
260 0679 3     ELSE
261 0680 4       ((.OPTIONS[0] AND NOT DISK_OPTIONS) NEQ 0
262 0681 4       OR (.OPTIONS[1] AND NOT DISK_OPTIONS2) NEQ 0 )
263 0682 3   END
264 0683 2 THEN ERR_EXIT (INITS_ILOOPT);
265 0684 2
266 0685 2 ! Now initialize the volume, depending on its type.
267 0686 2
268 0687 2
269 0688 2 IF .DEVICE_CHAR[DEV$V_SQD]
270 0689 2 THEN
271 0690 2   INIT_TAPE ()
272 0691 2 ELSE
273 0692 2   INIT_DISK ();
274 0693 2
275 0694 2 $DASSGN (CHAN = .CHANNEL);
276 0695 2 $DALLOC (DEVNAM = PHYS_NAME);
277 0696 2
278 0697 2 RETURN 1;
279 0698 2
280 0699 1 END;
```

! end of routine INIT_VOLUME

.TITLE INIVOL
.IDENT \V04-000\

```

        .PSECT $PLITS$,NOWRT,NOEXE,2

        0004 00000 P.AAA: .WORD 4
        0304 00002 .WORD 772
        00000000' 00004 .ADDRESS PROCESS_UIC
        00000000 00008 .LONG 0,0
        000C 00010 P.AAB: .WORD 12
        0202 00012 .WORD 514
        00000000G 00000000' 00014 .ADDRESS USER_STRING, USER_NAME
        00000000 0001C .LONG 0

        .PSECT $GLOBALS$,NOEXE,2

        00000 CLEANUP_FLAGS:: .BLKB 4
        00004 CHANNEL:: .BLKB 4
        00008 PROCESS_UIC:: .BLKB 4
        0000C DEVICE_CHAR:: .BLKB 4
        00080 DEVICE_CHAR2:: .BLKB 116
        000F4 PHYS_NAME:: .BLKB 116
        000FC NAME_BUFFER:: .BLKB 8
        0011C USER_STRING:: .BLKB 32
        00000074 00128 DEVCHAR_DESC:: .BLKB 12
        00000000' 0012C .LONG 116
        00000000 00130 DEVCHAR_DESC2:: .ADDRESS DEVICE_CHAR
        00000074 00130 DEVCHAR_DESC2:: .LONG 116
        00000000' 00134 .ADDRESS DEVICE_CHAR2

        NAMEBUF LEN== 32
        USERNAME LEN== 12
        .EXTRN INIT_OPTIONS, DEVICE_STRING
        .EXTRN USER_NAME, INIT_PARSE
        .EXTRN TRAN_LOGNAME, INIT_TAPE
        .EXTRN INIT_DISK, SYSSGETJPI
        .EXTRN SYSSALLOC, SYSSASSIGN
        .EXTRN SYSSGETCHN, SYSSDASSGN
        .EXTRN SYSSDALLOC

        .PSECT $CODE$,NOWRT,2

        59 0000G 03FC 00000 .ENTRY INIT_VOLUME, Save R2,R3,R4,R5,R6,R7,R8,R9 : 0533
        58 00000000G 00 9E 00002 MOVAB OPTIONS, R9
        57 00000000G 00 9E 00007 MOVAB SYSSGETJPI, R8
        57 00000000G 00 9E 0000E MOVAB LIBSTOP, R7
        56 0000' CF 9E 00015 MOVAB DEVICE_CHAR, R6
        5E 08 C2 0001A SUBL2 #8, SP
        60 015F CF DE 0001D MOVAL 17$, (FP)
        7E 7C 00022 CLRQ -(SP) : 0566
                                         : 0592

```


74	A6	00000000G	00	F8	7E	D4	000E4	CLRL	-(SP)			
			66	0074	A6	DD	000E6	PUSHL	CHANNEL			
					05	FB	000E9	CALLS	#5 SYSSGETCHN			
					8F	29	000F0	CMPC3	#116, DEVICE_CHAR, DEVICE_CHAR2		0658	
					05	12	000F7	BNEQ	7\$			
	J8	01	A6		06	E0	000F9	BBS	#6, DEVICE_CHAR+1, 8\$		0659	
			7E	01CC	8F	3C	000FE	7\$:	MOVZWL	#460, -(SP)		
			67		01	FB	00103	CALLS	#1, LIB\$STOP		0660	
	07	J2	A6		02	E0	00106	BBS	#2, DEVICE_CHAR+2, 9\$		0662	
			7E	84	8F	9A	0010B	MOVZBL	#132, -(SP)		0663	
			67		01	FB	0010F	CALLS	#1, LIB\$STOP			
	07	02	A6		03	E1	00112	BEC	#3, DEVICE_CHAR+2, 10\$		0665	
			7E	6C	8F	9A	00117	MOVZBL	#108, -(SP)		0666	
			67		01	FB	0011B	CALLS	#1, LIB\$STOP			
	13	F4	A6		02	88	0011E	BISB2	#2, CLEANUP FLAGS		0671	
		66			05	E1	00122	BBC	#5, DEVICE_CHAR, 11\$		0675	
		30FFDB98	8F		69	D3	00126	BITL	OPTIONS, #822074264		0677	
					1D	12	0012D	BNEQ	13\$			
		FFFFF835	8F	04	A9	D3	0012F	BITL	OPTIONS+4, #-1995		0678	
					11	11	00137	BRB	12\$			
		48000022	8F		69	D3	00139	11\$:	BITL	OPTIONS, #1207959586		0680
					0A	12	00140	BNEQ	13\$			
		FFFFF3CA	8F	04	A9	D3	00142	BITL	OPTIONS+4, #-3126		0681	
					09	13	0014A	BEQL	14\$			
				00758034	8F	DD	0014C	PUSHL	#7700532		0683	
	07		67		01	FB	00152	CALLS	#1, LIB\$STOP			
		0000G	66		05	E1	00155	BBC	#5, DEVICE_CHAR, 15\$		0688	
			CF		00	FB	00159	CALLS	#0, INIT_TAPE		0690	
		0000G	CF		05	11	0015E	BRB	16\$			
					00	FB	00160	CALLS	#0, INIT_DISK		0692	
		00000000G	00	F8	A6	DD	00165	16\$:	PUSHL	CHANNEL		
					01	FB	00168	CALLS	#1, SYSSDASSGN		0694	
		00000000G	00		7E	D4	0016F	CLRL	-(SP)		0695	
					C6	9F	00171	PUSHAB	PHYS_NAME			
		00000000G	50	00E8	02	FB	00175	CALLS	#2, SYSSDALLOC			
					01	D0	0017C	MOVL	#1, R0		0697	
					04	0017F	RET				0699	
					0000	00180	17\$:	.WORD	Save nothing		0566	
					7E	D4	00182	CLRL	-(SP)			
					5E	DD	00184	PUSHL	SP			
		0000V	7E	04	AC	7D	00186	MOVQ	4(AP), -(SP)			
			CF		03	FB	0018A	CALLS	#3, MAIN_HANDLER			
					04	0018F	RET					

: Routine Size: 400 bytes, Routine Base: \$CODE\$ + 0000

282 0700 1 ROUTINE MAIN_HANDLER (SIGNAL, MECHANISM) =
283 0701 1
284 0702 1 !++
285 0703 1
286 0704 1 FUNCTIONAL DESCRIPTION:
287 0705 1
288 0706 1 This routine is the main level condition handler for the INIT
289 0707 1 utility. It cleans up and returns the error status to the caller
290 0708 1 (the CLI).
291 0709 1
292 0710 1
293 0711 1 CALLING SEQUENCE:
294 0712 1 MAIN_HANDLER (ARG1, ARG2)
295 0713 1
296 0714 1 INPUT PARAMETERS:
297 0715 1 ARG1: address of signal array
298 0716 1 ARG2: address of mechanism array
299 0717 1
300 0718 1 IMPLICIT INPUTS:
301 0719 1 NONE
302 0720 1
303 0721 1 OUTPUT PARAMETERS:
304 0722 1 NONE
305 0723 1
306 0724 1 IMPLICIT OUTPUTS:
307 0725 1 NONE
308 0726 1
309 0727 1 ROUTINE VALUE:
310 0728 1 SSS\$_RESIGNAL
311 0729 1
312 0730 1 SIDE EFFECTS:
313 0731 1 cleanup done on unwind
314 0732 1
315 0733 1 --
316 0734 1
317 0735 2 BEGIN
318 0736 2
319 0737 2 MAP
320 0738 2 SIGNAL : REF BBLOCK, ! signal array
321 0739 2 MECHANISM : REF BBLOCK; ! mechanism array
322 0740 2
323 0741 2 EXTERNAL ROUTINE
324 0742 2 CLEAR_VALID: ! clear software volume valid
325 0743 2
326 0744 2
327 0745 2
328 0746 2 IF .BBLOCK [SIGNAL[CHF\$L_SIG_NAME], STSSV_FAC_NO] EQ 0
329 0747 2 OR .BBLOCK [SIGNAL[CHF\$L_SIG_NAME], STSSV_FAC_NO] EQ MOUNS_FACILITY
330 0748 2 THEN BBLOCK [SIGNAL[CHF\$L_SIG_NAME], STSSV_FAC_NO] = INITS_FACILITY;
331 0749 2
332 0750 2 IF .BBLOCK [SIGNAL[CHF\$L_SIG_NAME], STSSV_SEVERITY] EQ STSSK_SEVERE
333 0751 2 THEN
334 0752 3 BEGIN
335 0753 3
336 0754 3 IF .CHANNEL NEQ 0
337 0755 3 THEN
338 0756 4 BEGIN

! end of routine MAIN HANDLER

.EXTRN CLEAR VALID, SYSSQIOW
.EXTRN SYSSC~~M~~KRNL

: Routine Size: 132 bytes, Routine Base: \$CODE\$ + 0190

: 353 0771 1
: 354 0772 1 END
: 355 0773 0 ELUDOM

.EXTRN LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
\$GLOBALS	312	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$SPLITS	32	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	532	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
. ABS .	0	NOVEC, NOWRT, NORD, NOEXE, NOSHR, LCL, ABS, CON, NOPIC, ALIGN(0)

Library Statistics

File	----- Symbols -----			Pages Mapped	Processing Time
	Total	Loaded	Percent		
\$_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	36	0	1000	00:01.9

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:INIVOL/OBJ=OBJ\$:INIVOL MSRC\$:INIVOL/UPDATE=(ENH\$:INIVOL)

Size:	532 code + 344 data bytes
Run Time:	00:17.3
Elapsed Time:	00:41.1
Lines/CPU Min:	2688
Lexemes/CPU-Min:	39683
Memory Used:	157 pages
Compilation Complete	

0188 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

INPSMB
MAP

INSDIF
SOL

INPSMBMSG
LIS

RSXLBLOF
SOL

INSCREATE
LIS

INITIO
LIS

INSTAL
S

INSTALL
MAP

INSCMO
CLD

INSPREFIX
REQ

INPSMBCLD
CLD

INPSMB
LIS

INSOLDEMO
CLD

INSCMO
LIS

INITIO
LIS

ROHOME
LIS

INPSMB
LIS

INPSMBCLD
LIS